A Comparative Evaluation of State-of-the-Art Web Services Composition Testing Approaches

6th International Workshop on Automation of Software Test (AST 2011)
23rd May 2011, Honolulu, Hawaii

Hazlifah Mohd Rusli, Mazidah Puteh
Universiti Teknologi MARA, Malaysia
Suhaimi Ibrahim, Sayed Gholam Hassan Tabatabaei
Universiti Teknologi Malaysia, Malaysia

Content

- Motivation
- Related Work
- Web Services Composition
- Web Services Composition Testing
- Classification of Approaches
- Comparative Evaluation

MOTIVATION

- How are Web services composition tested?
- What are the issues being addressed by other researchers?
- What is being tested?
- What are the techniques used in generating tests for Web services composition testing?

Related Work

- Bozkurt et al. (2010), Metzger et al. (2010), Canfora et al. (2009)
  - Comprehensive and excellent surveys on Web services testing
- Bucchiarone et al. (2007)
  - Classify into orchestration and choreography aspect. Unit testing and integration testing
- Zakaria et al. (2009)
  - BPEL unit testing. No comparative evaluation of any of the approaches

Web Services Composition

- Required when a single Web service is unable to meet a client's requirement
- E.g – traveling
    - Book flight, reserve hotel, transportation, entertainment
- Executing each task one-by-one is time and effort consuming
- What is needed - Collection of services combined to achieve a user's request

Web Services Composition

- Composition schema or process definition defines business logic of a composite Web service
- Composition engine executes business logic by invoking services
- Orchestration schema specifies order of service invocation
- BPEL and OWL-S – two main languages for composing Web services
- Composition process: planning, discovery, selection and execution

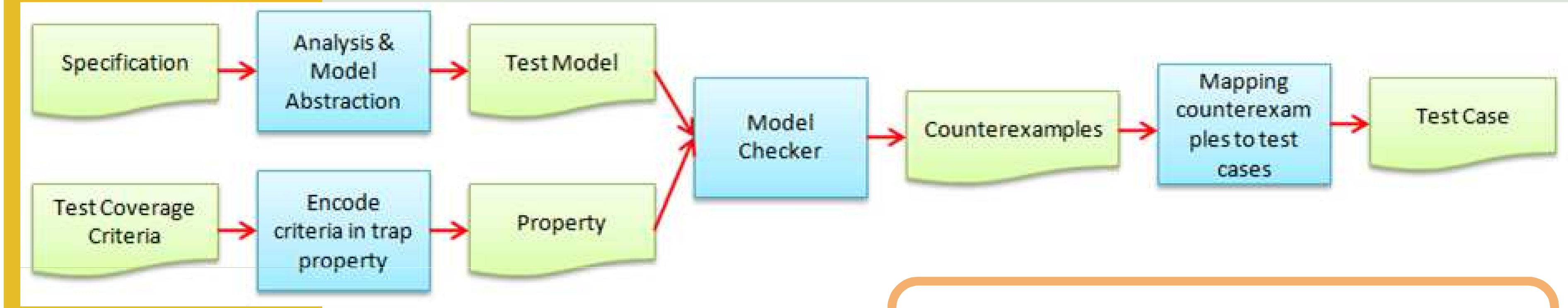
Web Services Composition Testing

- Static verification of Web services composition has been investigated for quite some time.
- Testing Web service composition is starting to garner interest.
- Static verification able to determine business process correctness in terms of deadlock freedom and reachability.
- Testing able to demonstrate the orchestration and sub-services used conforms to their publicized behavioral interfaces.

Classification Of WSC Testing Approaches

- Research papers used in the classification is based on the results of an earlier mapping study conducted between April to September 2010
- “involves a search of the literature to determine what sorts of studies addressing the systematic review question have been carried out, where they are published, in what databases they have been indexed, what sorts of outcomes they have assessed, and in which populations..” (Petticrew and Roberts, 2005)
- Classifications are based on test generation technology adopted by the researchers: Model checking, path analysis constraint solver approach, graph search algorithms and others

MODEL CHECKING APPROACH

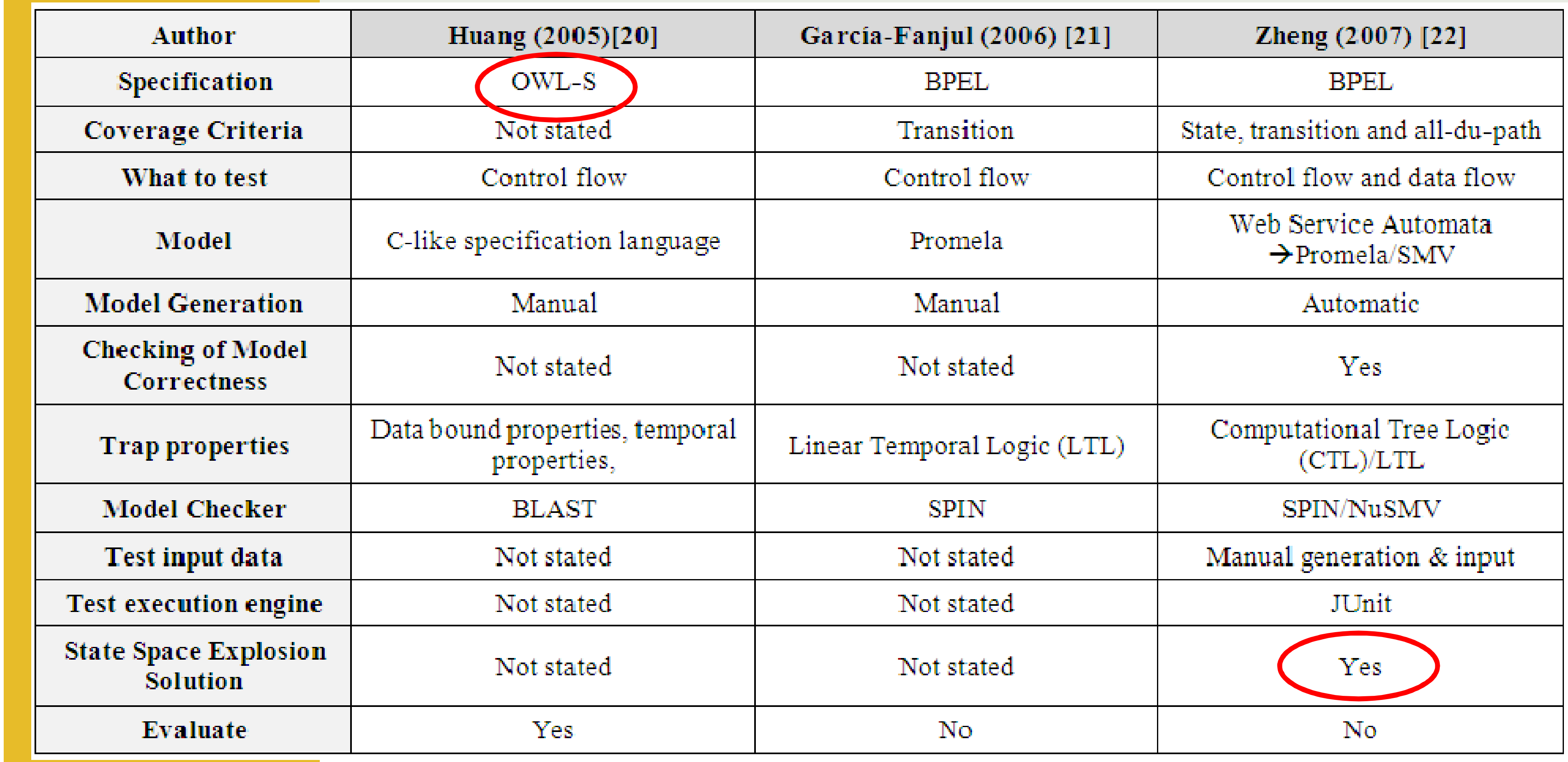

Specification
Analysis & Model Abstraction
Test Model
Test Coverage Criteria
Encode criteria in trap property
Property
Model Checker
Counterexamples
Mapping counterexamples to test cases
Test Case

***Trap property** - negation of original desired property*

***Counterexamples** - execution path that takes the finite state model from its initial state to a state where the violation occurs*

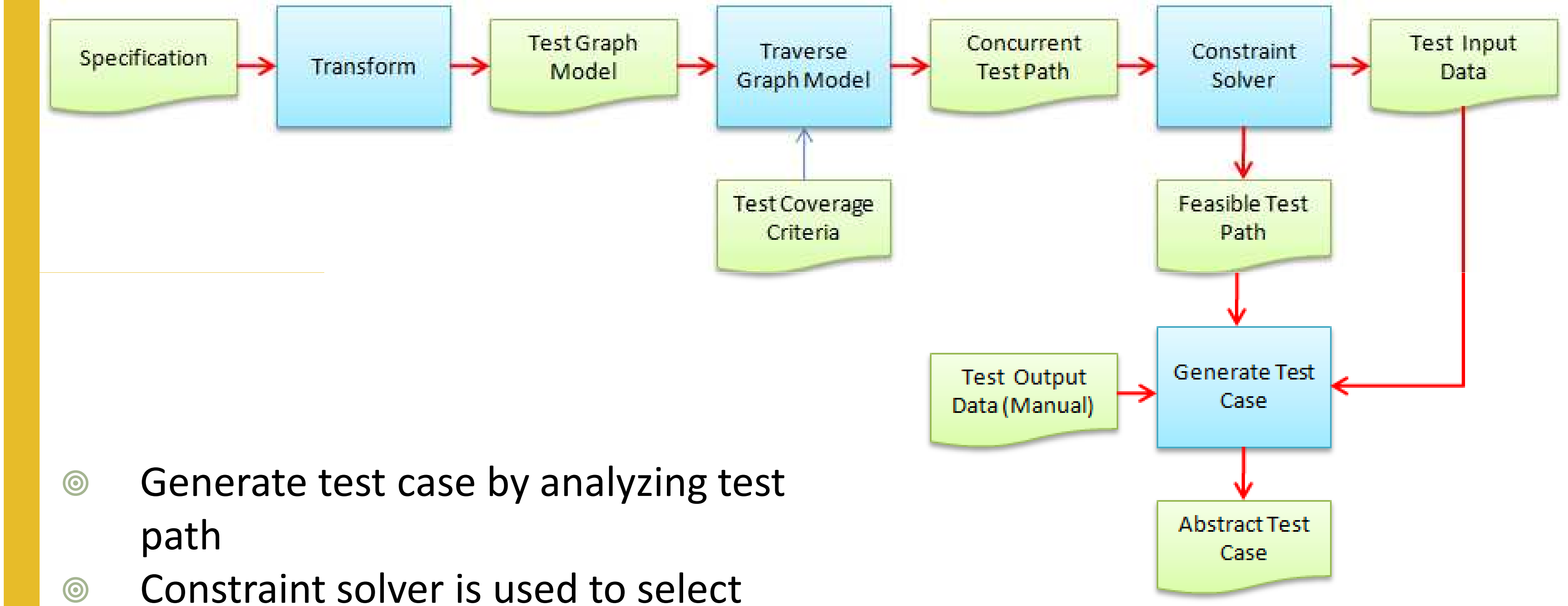
- Uses model checkers to generate test cases
- Advantage: Test output can be obtained from the model
- Disadvantage: State explosion problem

Model Checking Approach



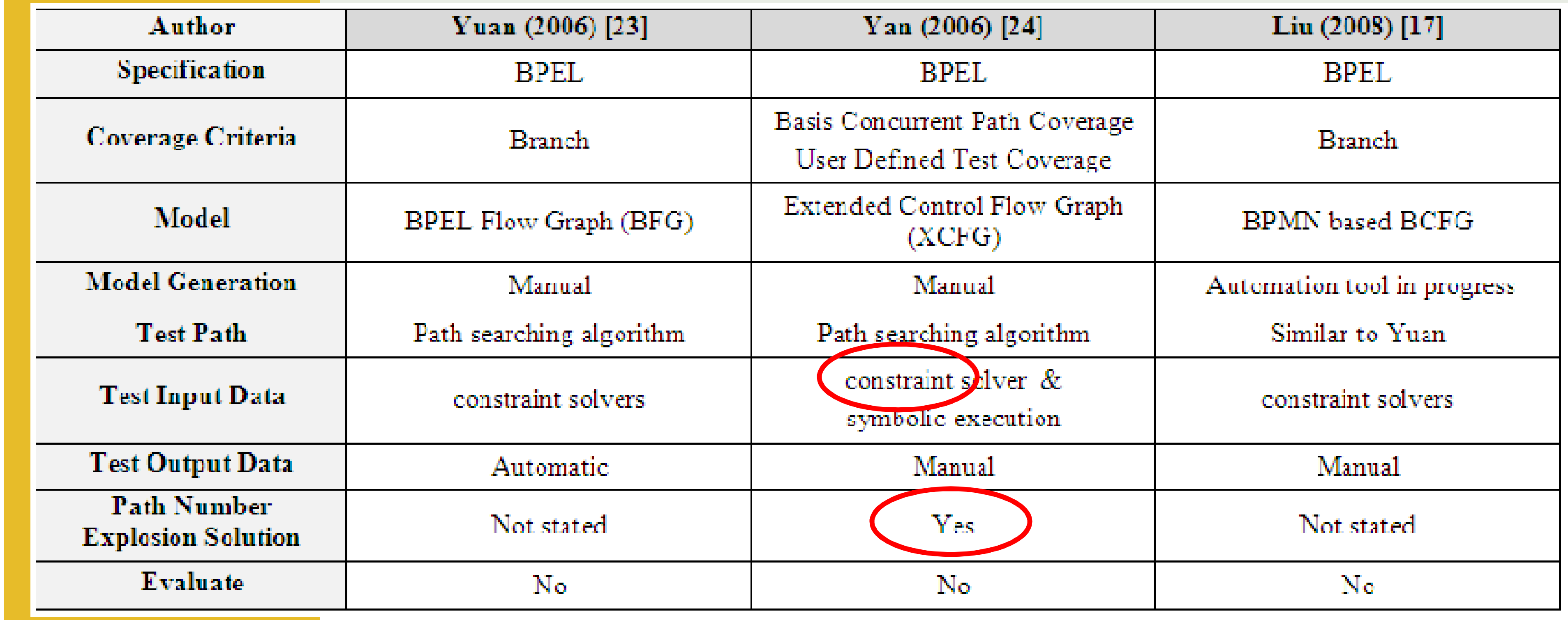
Author	Huang (2005)[20]	García-Fanjul (2006) [21]	Zheng (2007) [22]
Specification	OWL-S	BPEL	BPEL
Coverage Criteria	Not stated	Transition	State, transition and all-du-path
What to test	Control flow	Control flow	Control flow and data flow
Model	C-like specification language	Promela	Web Service Automata →Promela/SMV
Model Generation	Manual	Manual	Automatic
Checking of Model Correctness	Not stated	Not stated	Yes
Trap properties	Data bound properties, temporal properties,	Linear Temporal Logic (LTL)	Computational Tree Logic (CTL)/LTL
Model Checker	BLAST	SPIN	SPIN/NuSMV
Test input data	Not stated	Not stated	Manual generation & input
Test execution engine	Not stated	Not stated	JUnit
State Space Explosion Solution	Not stated	Not stated	Yes
Evaluate	Yes	No	No

PATH ANALYSIS CONSTRAINT SOLVER APPROACH



- Generate test case by analyzing test path
- Constraint solver is used to select feasible test path
- Advantage: Ability to select feasible test path and test input generation

PATH ANALYSIS CONSTRAINT SOLVER APPROACH

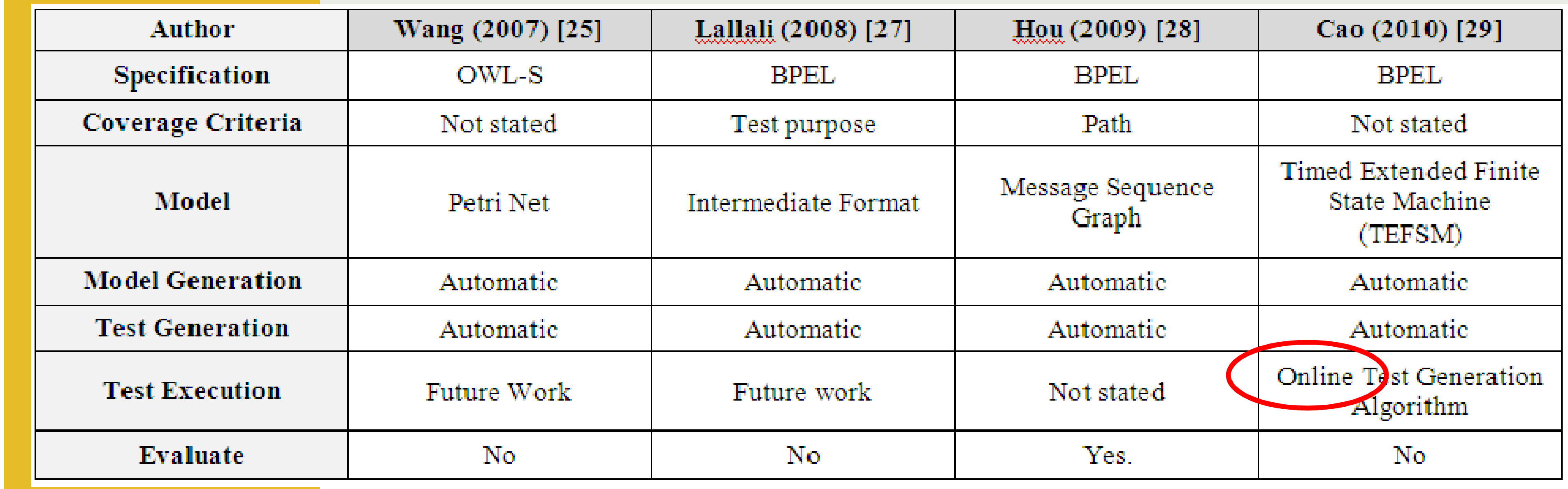


Author	Yuan (2006) [23]	Yan (2006) [24]	Liu (2008) [17]
Specification	BPEL	BPEL	BPEL
Coverage Criteria	Branch	Basis Concurrent Path Coverage User Defined Test Coverage	Branch
Model	BPEL Flow Graph (BFG)	Extended Control Flow Graph (XCFG)	BPMN based BCFG
Model Generation	Manual	Manual	Automation tool in progress
Test Path	Path searching algorithm	Path searching algorithm	Similar to Yuan
Test Input Data	constraint solvers	constraint solver & symbolic execution	constraint solvers
Test Output Data	Automatic	Manual	Manual
Path Number Explosion Solution	Not stated	Yes	Not stated
Evaluate	No	No	No

Graph Search Algorithm

- Similar to path analysis approach
- But does not make use of constraint solvers
- Steps:
  1. Transform composition specification into graph models
  2. Traverse graph in order to generate test cases

GRAPH SEARCH ALGORITHM



Author	Wang (2007) [25]	Lallali (2008) [27]	Hou (2009) [28]	Cao (2010) [29]
Specification	OWL-S	BPEL	BPEL	BPEL
Coverage Criteria	Not stated	Test purpose	Path	Not stated
Model	Petri Net	Intermediate Format	Message Sequence Graph	Timed Extended Finite State Machine (TEFSM)
Model Generation	Automatic	Automatic	Automatic	Automatic
Test Generation	Automatic	Automatic	Automatic	Automatic
Test Execution	Future Work	Future work	Not stated	Online Test Generation Algorithm
Evaluate	No	No	Yes.	No

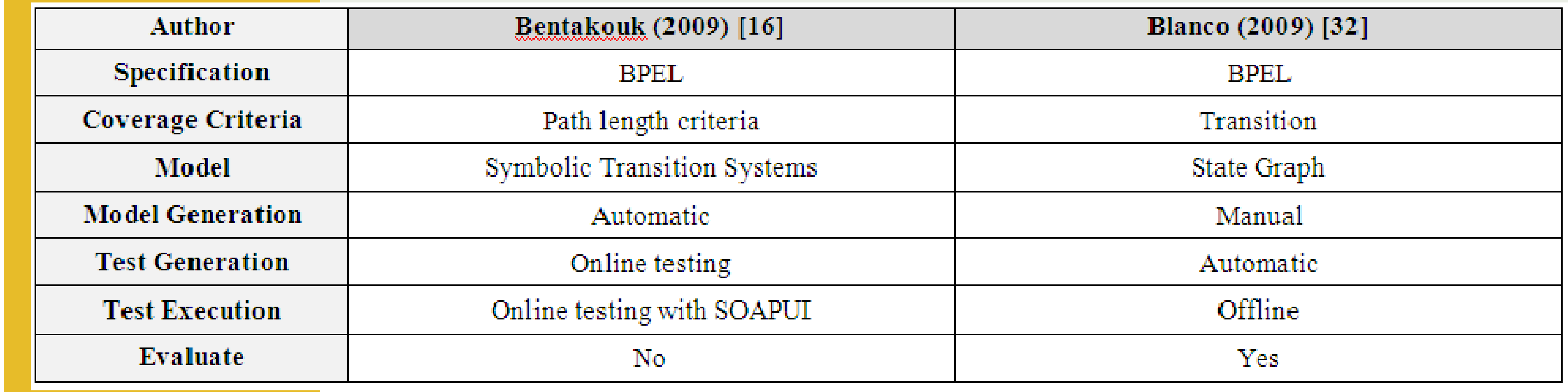
Other Approaches

- Symbolic Execution approach (Bentakouk, 2009)
  - Proposed to solve state explosion problem
  - Variable values are represented as symbolic values instead of concrete data
  - Steps:
    - Translate BPEL specification into Symbolic Transition Systems (STS)
    - Compute the Symbolic Execution Tree (SET) from STS
    - Generate symbolic test cases (test path) using SET and path coverage criterion
    - Online testing algorithm takes in the test path and SET to realize test case
    - Execute realized test case online

Other Approaches

- Metaheuristic search technique (Blanco, 2009)
  - Evolutionary method that works on a set of solutions of a problem to be solved
  - Aims to provide better solutions by combining existing solutions based on quality and diversity criteria
  - Business process is represented by a state graph and the goal of the approach is to allow all business process transitions to be covered

Other Approaches

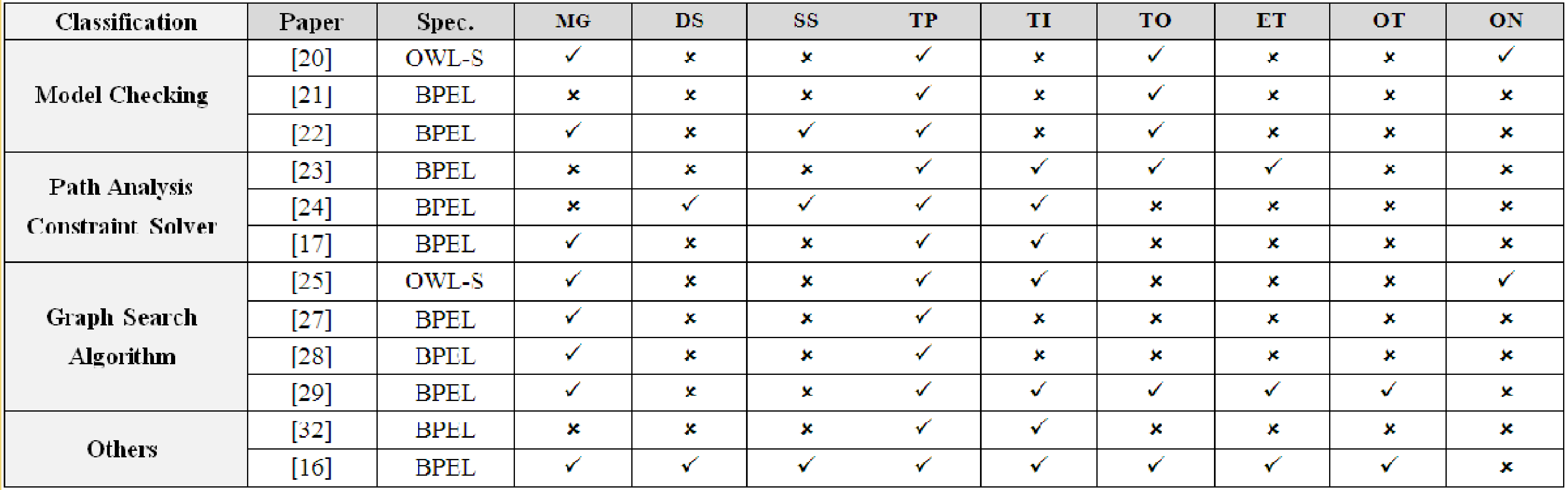


Author	Bentakouk (2009) [16]	Blanco (2009) [32]
Specification	BPEL	BPEL
Coverage Criteria	Path length criteria	Transition
Model	Symbolic Transition Systems	State Graph
Model Generation	Automatic	Manual
Test Generation	Online testing	Automatic
Test Execution	Online testing with SOAPUI	Offline
Evaluate	No	Yes

Comparative Evaluation Criteria

- Automatic Model Generation
- Different Composition Specification
- Avoid State Explosion/Path Number Explosion Problems
- Automatic Test Case Generation (Test Path, Test Input, Test Output)
- Executable Test Case
- Online Testing
- Ontology

Comparative Evaluation Result



Classification	Paper	Spec.	MG	DS	SS	TP	TI	TO	ET	OT	ON
Model Checking	[20]	OWL-S	✓	×	×	✓	×	✓	×	×	✓
	[21]	BPEL	×	×	×	✓	×	✓	×	×	×
	[22]	BPEL	✓	×	✓	✓	×	✓	×	×	×
Path Analysis Constraint Solver	[23]	BPEL	×	×	×	✓	✓	✓	✓	×	×
	[24]	BPEL	×	✓	✓	✓	✓	×	×	×	×
	[17]	BPEL	✓	×	×	✓	✓	×	×	×	×
Graph Search Algorithm	[25]	OWL-S	✓	×	×	✓	✓	×	×	×	✓
	[27]	BPEL	✓	×	×	✓	×	×	×	×	×
	[28]	BPEL	✓	×	×	✓	×	×	×	×	×
	[29]	BPEL	✓	×	×	✓	✓	✓	✓	✓	×
Others	[32]	BPEL	×	×	×	✓	✓	×	×	×	×
	[16]	BPEL	✓	✓	✓	✓	✓	✓	✓	✓	×

MG - Automatic Model Generation

DS - Different Composition Specification

SS - Avoid State Explosion/Path Number Explosion Problems

TP – Test Path

TI – Test Input

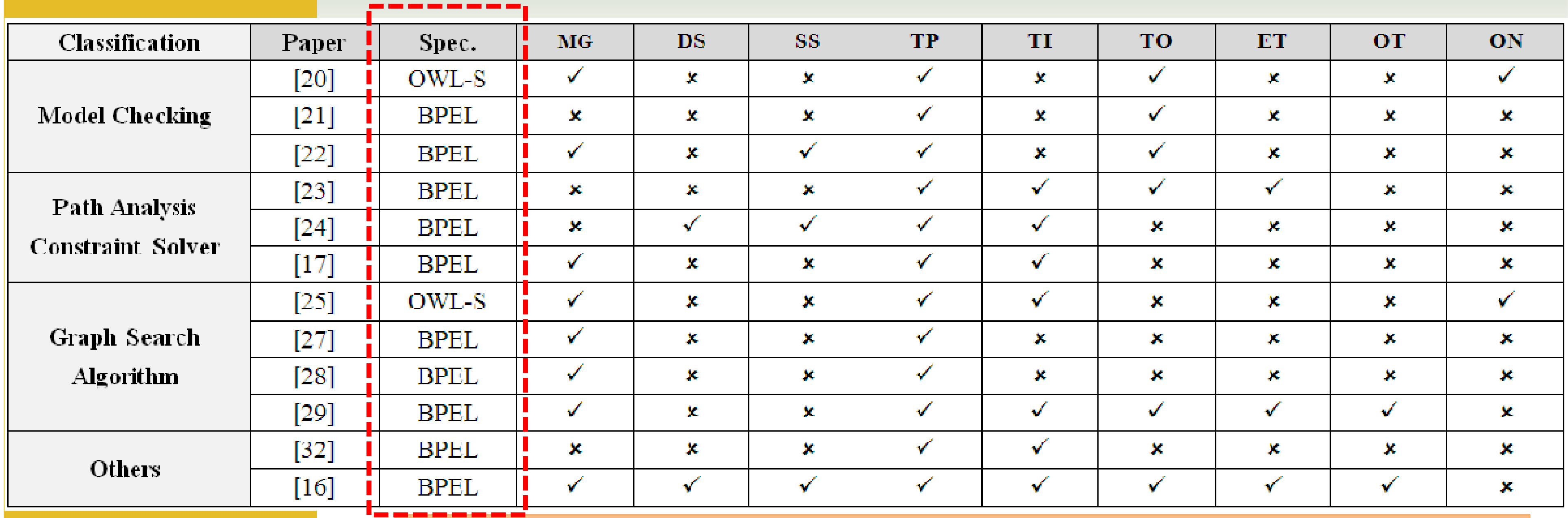
TO – Test Output

ET - Executable Test Case

OT - Online Testing

ON - Ontology

Comparative Evaluation



Classification	Paper	Spec.	MG	DS	SS	TP	TI	TO	ET	OT	ON
Model Checking	[20]	OWL-S	✓	×	×	✓	×	✓	×	×	✓
	[21]	BPEL	×	×	×	✓	×	✓	×	×	×
	[22]	BPEL	✓	×	✓	✓	×	✓	×	×	×
Path Analysis Constraint Solver	[23]	BPEL	×	×	×	✓	✓	✓	✓	×	×
	[24]	BPEL	×	✓	✓	✓	✓	×	×	×	×
	[17]	BPEL	✓	×	×	✓	✓	×	×	×	×
Graph Search Algorithm	[25]	OWL-S	✓	×	×	✓	✓	×	×	×	✓
	[27]	BPEL	✓	×	×	✓	×	×	×	×	×
	[28]	BPEL	✓	×	×	✓	×	×	×	×	×
	[29]	BPEL	✓	×	×	✓	✓	✓	✓	✓	×
Others	[32]	BPEL	×	×	×	✓	✓	×	×	×	×
	[16]	BPEL	✓	✓	✓	✓	✓	✓	✓	✓	×

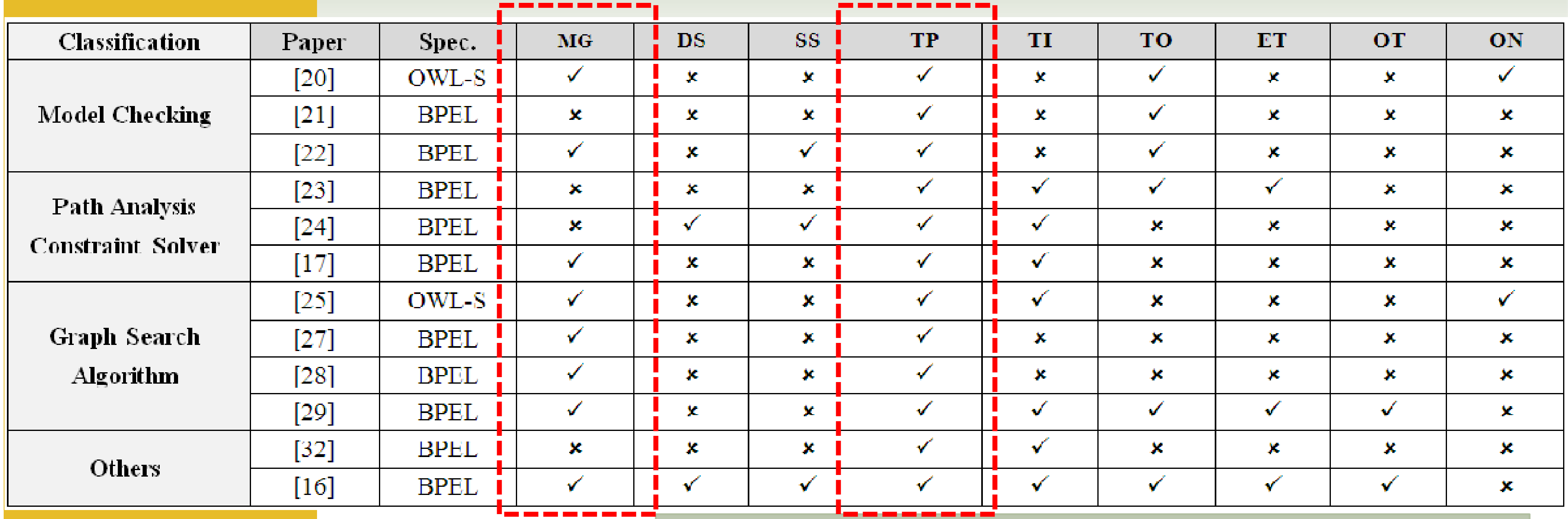
Eight (8) out of the twelve (12) approaches worked on syntax based Web services composition testing (BPEL). Only two approaches worked on semantic Web services (OWL-S)

MG - Automatic Model Generation
DS - Different Composition Specification
SS - Avoid State Explosion/Path Number Explosion Problems

TP – Test Path
TI – Test Input
TO – Test Output

ET - Executable Test Case
OT - Online Testing
ON - Ontology

Comparative Evaluation Result

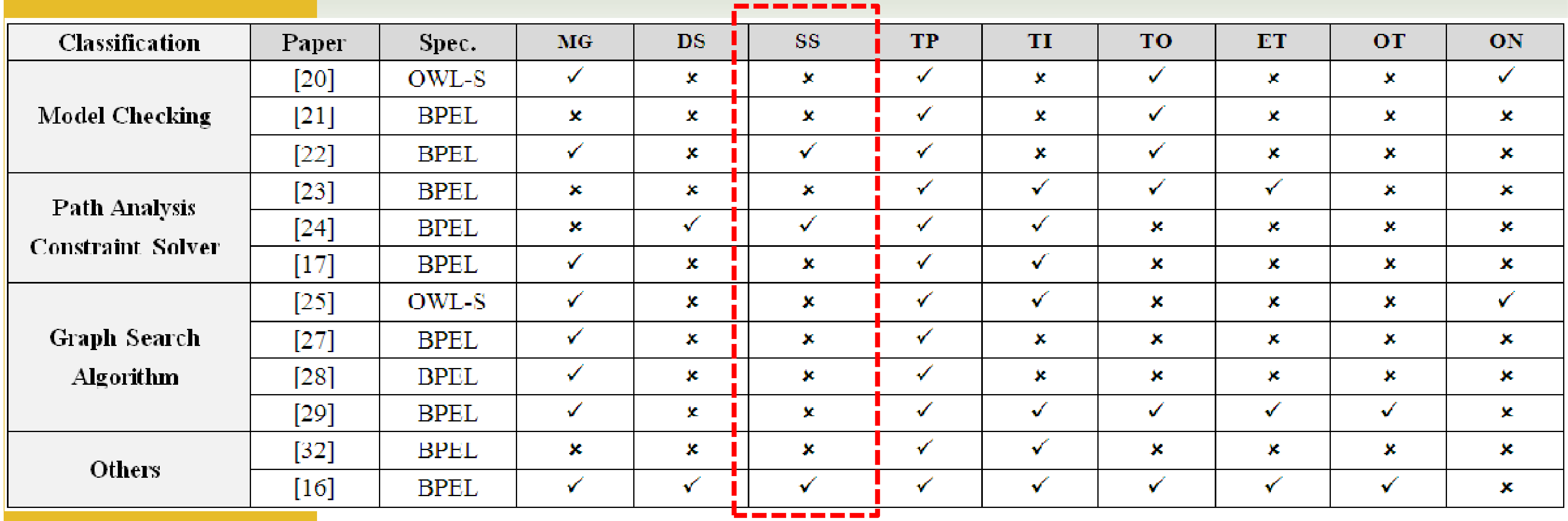


Classification	Paper	Spec.	MG	DS	SS	TP	TI	TO	ET	OT	ON
Model Checking	[20]	OWL-S	✓	×	×	✓	×	✓	×	×	✓
	[21]	BPEL	×	×	×	✓	×	✓	×	×	×
	[22]	BPEL	✓	×	✓	✓	×	✓	×	×	×
Path Analysis Constraint Solver	[23]	BPEL	×	×	×	✓	✓	✓	✓	×	×
	[24]	BPEL	×	✓	✓	✓	✓	×	×	×	×
	[17]	BPEL	✓	×	×	✓	✓	×	×	×	×
Graph Search Algorithm	[25]	OWL-S	✓	×	×	✓	✓	×	×	×	✓
	[27]	BPEL	✓	×	×	✓	×	×	×	×	×
	[28]	BPEL	✓	×	×	✓	×	×	×	×	×
	[29]	BPEL	✓	×	×	✓	✓	✓	✓	✓	×
Others	[32]	BPEL	×	×	×	✓	✓	×	×	×	×
	[16]	BPEL	✓	✓	✓	✓	✓	✓	✓	✓	×

Most approaches were able to automatically translate the composition specification into test models and to automatically generate test path

MG - Automatic Model Generation	**TP** – Test Path	**ET** - Executable Test Case
DS - Different Composition Specification	**TI** – Test Input	**OT** - Online Testing
SS - Avoid State Explosion/Path Number Explosion Problems	**TO** – Test Output	**ON** - Ontology

Comparative Evaluation Result



Classification	Paper	Spec.	MG	DS	SS	TP	TI	TO	ET	OT	ON
Model Checking	[20]	OWL-S	✓	×	×	✓	×	✓	×	×	✓
	[21]	BPEL	×	×	×	✓	×	✓	×	×	×
	[22]	BPEL	✓	×	✓	✓	×	✓	×	×	×
Path Analysis Constraint Solver	[23]	BPEL	×	×	×	✓	✓	✓	✓	×	×
	[24]	BPEL	×	✓	✓	✓	✓	×	×	×	×
	[17]	BPEL	✓	×	×	✓	✓	×	×	×	×
Graph Search Algorithm	[25]	OWL-S	✓	×	×	✓	✓	×	×	×	✓
	[27]	BPEL	✓	×	×	✓	×	×	×	×	×
	[28]	BPEL	✓	×	×	✓	×	×	×	×	×
	[29]	BPEL	✓	×	×	✓	✓	✓	✓	✓	×
Others	[32]	BPEL	×	×	×	✓	✓	×	×	×	×
	[16]	BPEL	✓	✓	✓	✓	✓	✓	✓	✓	×

Only three (3) approaches used techniques to avoid state explosion problems

MG - Automatic Model Generation

DS - Different Composition Specification

SS - Avoid State Explosion/Path Number Explosion Problems

TP – Test Path

TI – Test Input

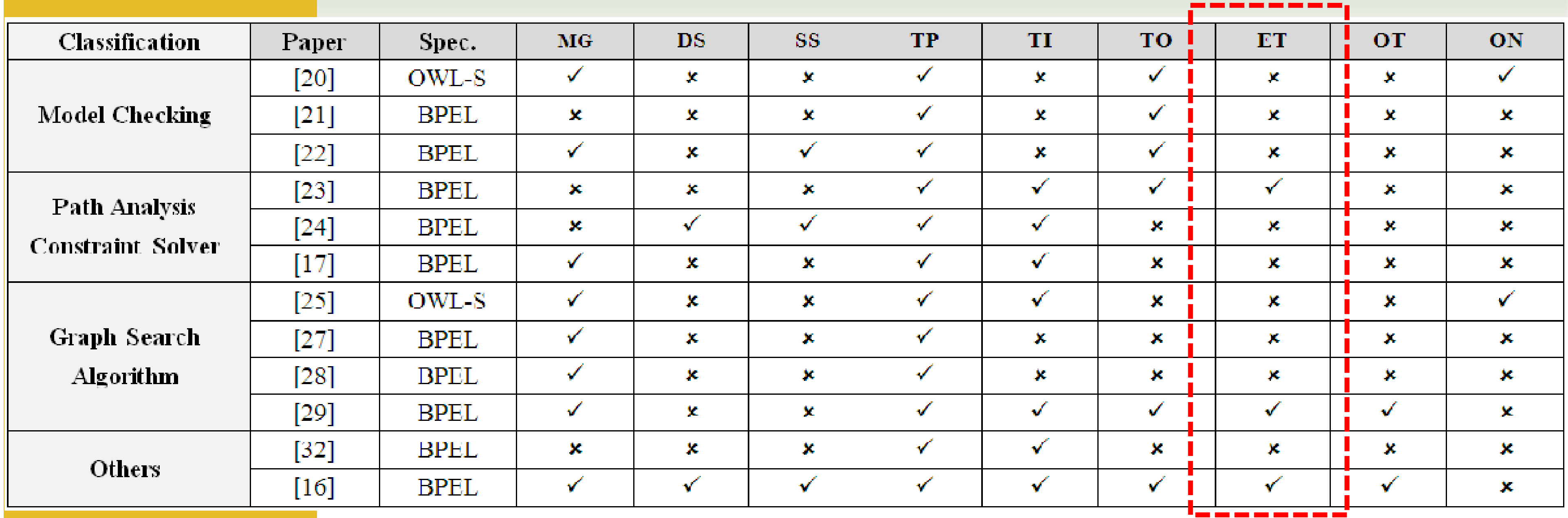
TO – Test Output

ET - Executable Test Case

OT - Online Testing

ON - Ontology

Comparative Evaluation Result



Classification	Paper	Spec.	MG	DS	SS	TP	TI	TO	ET	OT	ON
Model Checking	[20]	OWL-S	✓	×	×	✓	×	✓	×	×	✓
	[21]	BPEL	×	×	×	✓	×	✓	×	×	×
	[22]	BPEL	✓	×	✓	✓	×	✓	×	×	×
Path Analysis Constraint Solver	[23]	BPEL	×	×	×	✓	✓	✓	✓	×	×
	[24]	BPEL	×	✓	✓	✓	✓	×	×	×	×
	[17]	BPEL	✓	×	×	✓	✓	×	×	×	×
Graph Search Algorithm	[25]	OWL-S	✓	×	×	✓	✓	×	×	×	✓
	[27]	BPEL	✓	×	×	✓	×	×	×	×	×
	[28]	BPEL	✓	×	×	✓	×	×	×	×	×
	[29]	BPEL	✓	×	×	✓	✓	✓	✓	✓	×
Others	[32]	BPEL	×	×	×	✓	✓	×	×	×	×
	[16]	BPEL	✓	✓	✓	✓	✓	✓	✓	✓	×

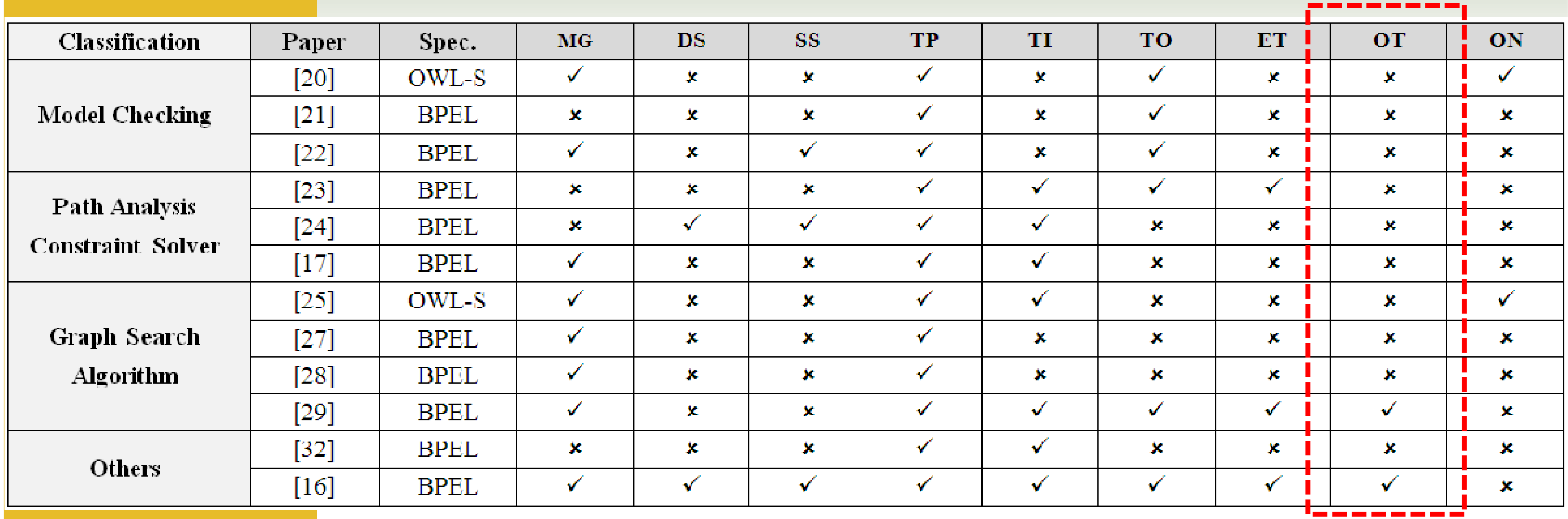
Only two (2) approaches generated executable test cases

MG - Automatic Model Generation
DS - Different Composition Specification
SS - Avoid State Explosion/Path Number Explosion Problems

TP – Test Path
TI – Test Input
TO – Test Output

ET - Executable Test Case
OT - Online Testing
ON - Ontology

Comparative Evaluation Result



Classification	Paper	Spec.	MG	DS	SS	TP	TI	TO	ET	OT	ON
Model Checking	[20]	OWL-S	✓	×	×	✓	×	✓	×	×	✓
	[21]	BPEL	×	×	×	✓	×	✓	×	×	×
	[22]	BPEL	✓	×	✓	✓	×	✓	×	×	×
Path Analysis Constraint Solver	[23]	BPEL	×	×	×	✓	✓	✓	✓	×	×
	[24]	BPEL	×	✓	✓	✓	✓	×	×	×	×
	[17]	BPEL	✓	×	×	✓	✓	×	×	×	×
Graph Search Algorithm	[25]	OWL-S	✓	×	×	✓	✓	×	×	×	✓
	[27]	BPEL	✓	×	×	✓	×	×	×	×	×
	[28]	BPEL	✓	×	×	✓	×	×	×	×	×
	[29]	BPEL	✓	×	×	✓	✓	✓	✓	✓	×
Others	[32]	BPEL	×	×	×	✓	✓	×	×	×	×
	[16]	BPEL	✓	✓	✓	✓	✓	✓	✓	✓	×

Only two (2) approaches attempted online testing

MG - Automatic Model Generation

DS - Different Composition Specification

SS - Avoid State Explosion/Path Number Explosion Problems

TP – Test Path

TI – Test Input

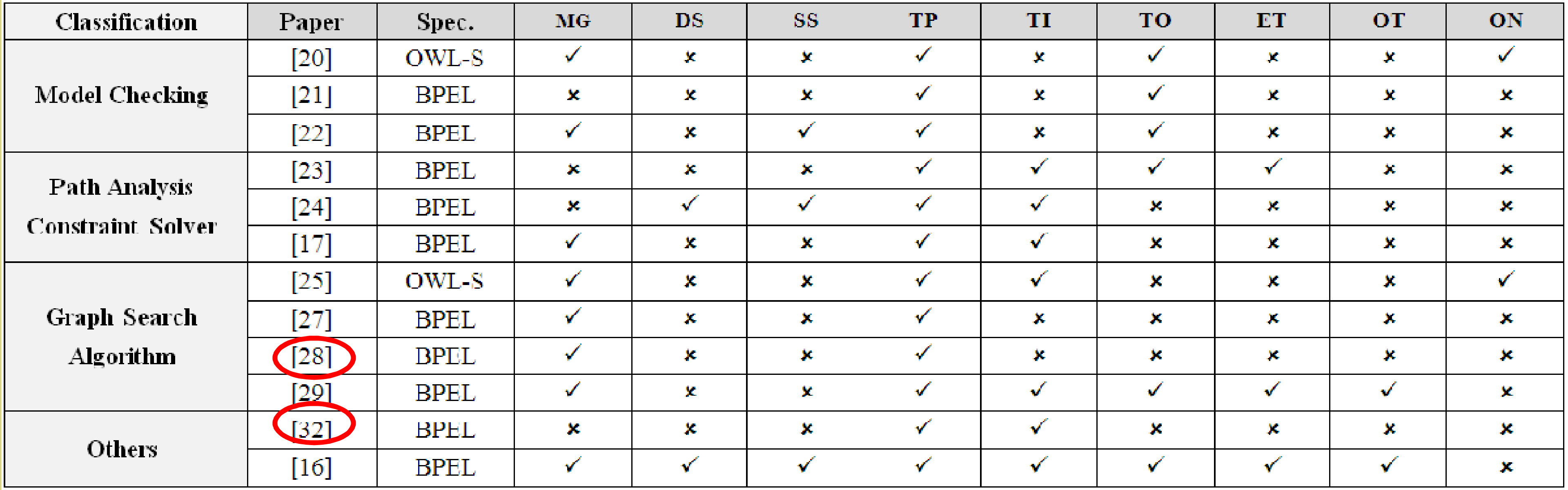
TO – Test Output

ET - Executable Test Case

OT - Online Testing

ON - Ontology

Comparative Evaluation Result



Classification	Paper	Spec.	MG	DS	SS	TP	TI	TO	ET	OT	ON
Model Checking	[20]	OWL-S	✓	×	×	✓	×	✓	×	×	✓
	[21]	BPEL	×	×	×	✓	×	✓	×	×	×
	[22]	BPEL	✓	×	✓	✓	×	✓	×	×	×
Path Analysis Constraint Solver	[23]	BPEL	×	×	×	✓	✓	✓	✓	×	×
	[24]	BPEL	×	✓	✓	✓	✓	×	×	×	×
	[17]	BPEL	✓	×	×	✓	✓	×	×	×	×
Graph Search Algorithm	[25]	OWL-S	✓	×	×	✓	✓	×	×	×	✓
	[27]	BPEL	✓	×	×	✓	×	×	×	×	×
	[28]	BPEL	✓	×	×	✓	×	×	×	×	×
	[29]	BPEL	✓	×	×	✓	✓	✓	✓	✓	×
Others	[32]	BPEL	×	×	×	✓	✓	×	×	×	×
	[16]	BPEL	✓	✓	✓	✓	✓	✓	✓	✓	×

Only two (2) approaches evaluated their approach based on benchmarks such as test coverage percentage, number of test cases generated, time it took to generate test cases, ratio of ineffective test cases and ratio of fault exposure

Conclusion

- Presented motivation and related work
- Presented an introduction to WSC and the need for testing WSC
- Provided an overview and evaluation of current approaches to WSC testing
- Presented most prominent approach and classified them into several categories
- Introduced criteria to evaluate them
- Presented the evaluation result

THANK YOU

Mapping Study on WSCT

- Research objectives
- Sources
- Search Criteria
- 42 research papers were selected and classified into test generation, framework, mutation testing, regression testing, survey and others